Name : Sujan Kumar Mitra
Roll : 09
Dept : CSE
Sem : 5th.

OS Assignment 1

1. What do you mean by Operating System? What are the important functions of an operating System?

Ans: An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

The important functions of an operating system are:

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Job accounting
- Error detecting aids
- Coordination between other software and users.

2. What do you mean by Multitasking?

Ans: Multi-tasking refers to the term where multiple jobs are executed by the CPU simultaneously by switching between them. Switches occur so frequently that the user may interact with each program while it is running. Multitasking OS are also referred as Time-Sharing OS.

3. What do you mean by Multi~~tasking~~ programming?

Ans: When two or more programs are residing in memory at the same time, then sharing the processor is referred to the multiprogramming. Multiprogramming assumes single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

4. What do you mean by SPOOLING?

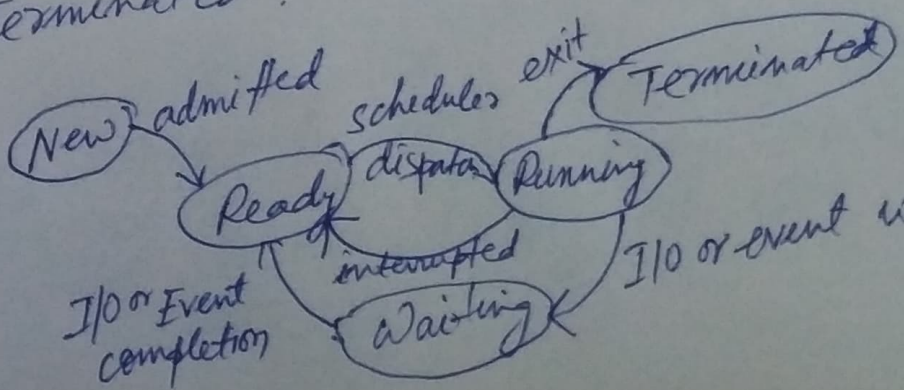
Ans. Spooling stands for Simultaneous Peripheral operations Online. Spooling refers to putting data of various I/O jobs in buffer. This buffer is a special area in memory or disk which is accessible to I/O devices. Spooling is done to bridge the gap of access rates of I/O devices and CPU.

5. What is Process? What are the state of a process and describe it with suitable picture.

Ans: A process is defined as an entity which represents the basic unit of work to be implemented in the system. In simpler words, a process is a program in execution.

Process State: As a process executes, it changes state. The state of a process is defined as the current activity of the process. The states are:

1. New: The process is being created
2. Ready: The process is ready to be executed and is waiting to be assigned to a processor.
3. Running: Process instructions are being executed.
4. Waiting: Process is waiting for some event to occur (like I/O completion)
5. Terminated: Process has finished execution



6. What is Process Control Block (PCB)? (Describe all information it contain)

Ans:

Each process is represented in the operating system by a process control block (PCB) also called task control block. PCB is a data structure used by operating system. OS groups all the information that needs about particular ~~block~~ process.

PCB contains many pieces of info associated with a specific process.

- Pointer : Pointer points to another PCB. It is used for maintaining scheduling list.
- Process state : Process state may be new, ready, running or waiting.
- Program Counter : It indicates address of next instruction to be executed.
- CPU Registers : They include general purpose registers, stack pointer, index registers and accumulators.
- Memory Management Info : They may include base and limit register, page table or segment table depending on OS.

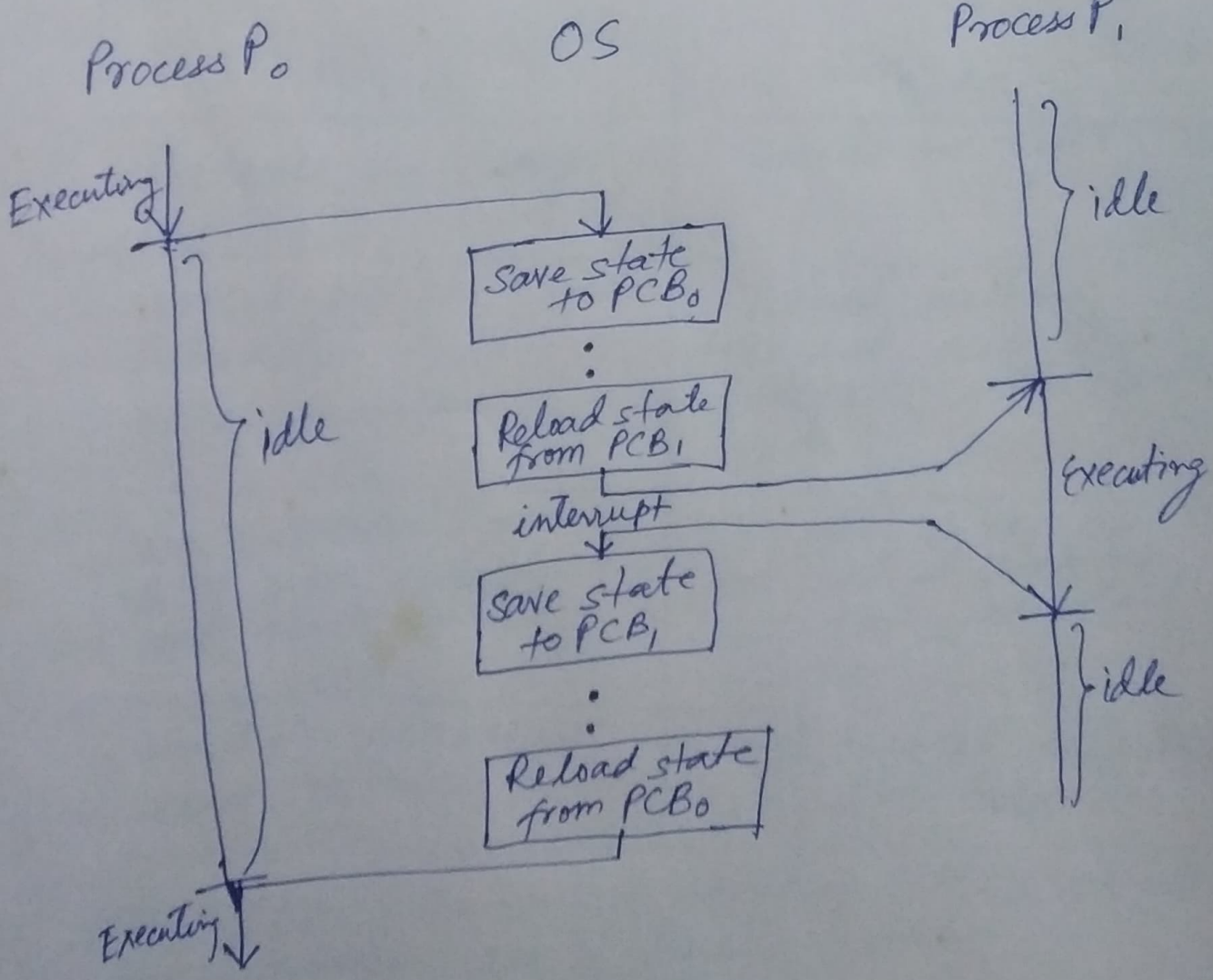
Accounting Info : This information includes the amount of CPU used, time limits, job or process numbers etc.

7. Write the difference between Long Term, Short Term and Medium Term Schedulers.

Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
It is a job scheduler	It is a CPU scheduler	It is process swapping scheduler
Speed is lesser than short term scheduler	Speed is fastes among other two	Speed is between both short term and long term
It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
It is almost absent in time sharing systems	It is minimal in time sharing systems	It is part of time sharing systems
It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

8. What do you mean by Context Switch with proper example?

Ans: A context switch is the mechanism to store and restore the state or context of a CPU in PCB so that a process execution can be resumed from same point at a later time. This technique enables multiple processes to share a single CPU.



9. What is Thread? Write the difference between Process and Thread? What are the benefits of Thread?

Ans: A thread is a flow of execution through the process code, with its own program counter, system registers and stack. A thread is also called light weigh process. Threads provide a way to improve application performance by parallelism.

Process	Thread
Processes are heavy weight	Threads are light weight
Process switching needs interaction with operating system	Thread switching does not need to interact with operating system
Each process have their own code, data and stack area.	Threads share the code and data area but have individual stacks.
Multiple processes use more resources	Multi threaded processes use fewer resources.

Advantages: The context switching time of threads are minimum. Use of threads provides concurrency within processes.

10. What is User Level Threads and Kernel Level Threads? Write the difference between user level and Kernel level threads.

User Level Threads:

The user level threads are implemented by users and the kernel is not aware of the existence of these threads. It handles them as if they were single threaded processes. They are represented by program counter, stack, registers and a small PCB. Kernel is not involved in synchronization for user-level threads.

Kernel Level Threads:

Kernel level threads are handled by the OS directly and thread management is done by the kernel. The context information for the process as well as the process threads is all managed by kernel.

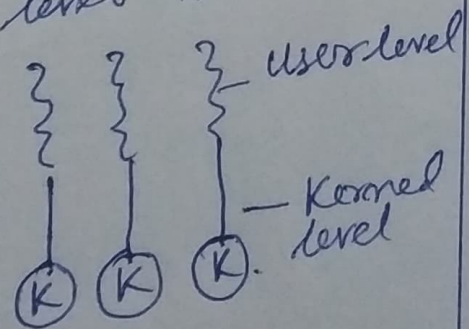
User Level Thread	Kernel Level Thread
User threads are implemented by users.	Kernel threads are implemented by ~~kernels~~ OS
Context switch time is less	Context switch time is more
Context switch requires no hardware support	Hardware support is needed.
Implementation is easy	Implementation is difficult.

11. What are the relationships in Multithreading models (between User Level Threads and Kernel Level Threads)? With proper diagram.

Ans:

One to One Model

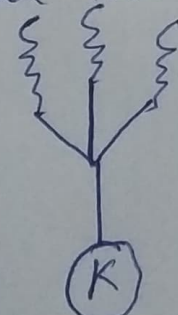
One user level thread is mapped to one kernel level thread



Ex- Windows

Many to One Model

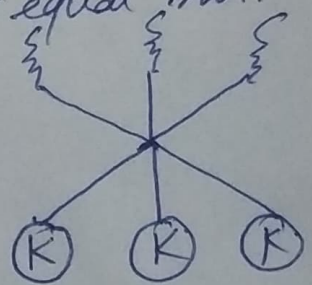
Many user level threads are mapped to one kernel level thread.



Ex- Green Solaris

Many to Many Model

Many user level threads multiplexes to the kernel threads of smaller or equal nos.



Ex - Java Threads

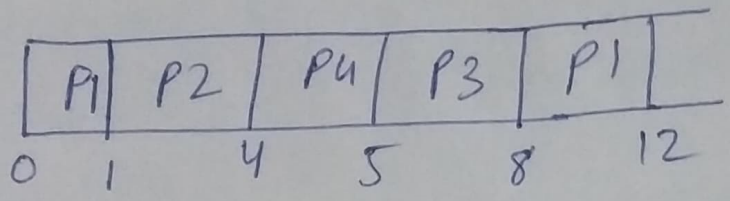
12. Consider the following 4 processes with length of CPU burst time given in milliseconds
Find out the i) Avg. waiting time
ii) Average turn around time iii) Average response time. for preemptive SJF scheduling.

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	3
P4	4	1

Ans:

Process	AT	BT	CT	TAT	WT	RT
P1	0	5	12	12	7	0
P2	1	3	4	3	0	0
P3	2	3	8	6	3	3
P4	4	1	5	1	0	0

Gnatt Chart:



Avg. turn around time: (12+3+6+1)/4

= 5.5

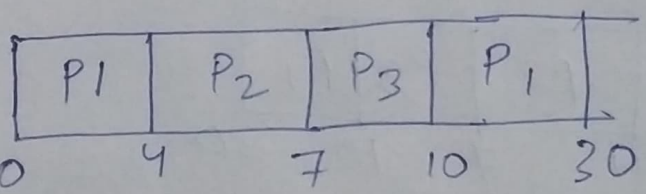
Avg. waiting time: (7+0+3+0)/4 = 2.5

Avg. response time: (0+0+3+0)/4 = 0.75

13. Consider the following 3 processes with lengths of CPU burst time given in milliseconds. Find out Average waiting time for RR Scheduling (if Time quantum is 4ms)

Process	Busst Time
P1	24
P2	3
P3	3

Ans: Gnatt Chart



Process	BT	CT	WT
P1	24	30	6
P2	3	7	4
P3	3	10	7

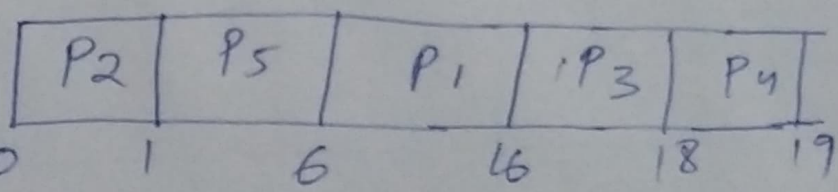
Avg. waiting time = 17/3
= 5.67

14. Consider the following processes with the length of CPU burst time given in milliseconds. Find out average waiting time for priority scheduling (smallest integer = highest priority and all processes arrive at time 0)

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Ans:

Gnatt chart:



Avg. waiting time = (1+6+16+18)/5

= 8.2

15. What is Semaphore? what are the operations on it? Determine the solution of 'Readers-Writers Problem' using Semaphore.

Ans: A semaphore is an integer variable that ~~apart~~ are used to solve the critical section problem by using two atomic operations wait() and signal().

Wait	Signal
This operation is used to acquire the lock	This operation is used to release the lock

wait(S) {
while (S<=0);
S--;
}
	signal(S) {
	++S;
	}

Solution of Readers Writers Problem:

Shared data structures:

```
Semaphore rw_mutex = 1,
Semaphore mutex = 1;
int read_count = 0;
```

Writer code:

```
while (true) {
    wait (rw_mutex);
    /* writing is performed */
    signal (rw_mutex);
}
```

Reader code:

```
while (true) {
    wait (mutex);
    read_count ++;
    if (read_count == 1)
        wait (rw_mutex);
    signal (mutex);
    /* reading is performed */
    wait (mutex);
    read_count --;
    if (read_count == 0)
        signal (rw_mutex);
    signal (mutex);
}
```

16. Determine the solution of Dining Philosopher problem using semaphore.

Ans:

```
semaphore chopstick[5];

while (true) {
    wait (chopstick[i]);
    wait (chopstick[(i+1)%5]);

    /* eat for a while */
    signal (chopstick[i]);
    signal (chopstick[(i+1)%5]);
```

To avoid deadlock, do either one of the following:

1) Allow at most four philosophers
2) Allow a philosopher to pick up both chopsticks or none.
3) Use asymmetric solution - odd-numbered philosopher picks left chopstick then right and even-numbered philosopher picks right chopstick, then left.

17. Determine Producer Consumer solution using semaphore.

Ans:

```
semaphore empty = 1;
semphore  full  = 1;
semaphore mutex = 1;

Producer:
    while (true) {
      wait (empty);
      wait (mutex);
      /* Produce an item to buffer */.
      signal (mutex);
      signal (full);
    }
Consumer:
    while (true) {
      wait (full);
      wait (mutex);
      /* consume an item from buffer */.
      signal (mutex);
      signal (empty);
    }
```